日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: ２０００年 ３月２２日

出 願 番 号
Application Number: 特願２０００－０８４７０６

出 願 人
Applicant (s): 株式会社日立製作所

２０００年 ６月２９日

特 許 庁 長 官
Commissioner,
Patent Office 近 藤 隆 彦

出証番号 出証特２０００－３０５２５１２

【書類名】　特許願
【整理番号】　K99009331
【提出日】　平成12年 3月22日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 12/00
【請求項の数】　11
【発明者】
　【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内
　【氏名】　荒井　正人
【発明者】
　【住所又は居所】　愛知県尾張旭市晴丘町池上１番地　株式会社日立製作所情報機器事業部内
　【氏名】　梶浦　敏範
【特許出願人】
　【識別番号】　000005108
　【氏名又は名称】　株式会社日立製作所
【代理人】
　【識別番号】　100075096
　【弁理士】
　【氏名又は名称】　作田　康夫
【手数料の表示】
　【予納台帳番号】　013088
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　アクセス制御システム

【特許請求の範囲】

【請求項１】

ファイルへの入出力手段と、

前記ファイルを格納する記憶手段と、

前記ファイルへのアクセス制御ポリシーを記述したポリシーファイルと、

当該ファイルアクセス要求の正当性を前記に基づき判定するアクセス制御手段と、

前記入出力手段を用いたファイルアクセス要求の発行を監視し、発行されたファイルアクセス要求を前記アクセス制御手段に伝え、正当性判定結果を前記アクセス制御手段から受信する監視手段と、を備え、

前記アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を前記ポリシーファイルに記し、

前記監視手段は、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を用いて、前記発行されたファイルアクセス要求を伝えることを特徴とするアクセス制御システム。

【請求項２】

請求項１記載のアクセス制御システムであって、

前記アクセス制御ポリシーは、前記ファイルに対して禁止されたアクセスタイプと、該禁止されたアクセスが発生した場合にアクセス要求発行元に返すべきエラーコードと、例外として該アクセス要求を許可されたアクセス実行手段とアクセス要求発行元とを特定する情報とからなることを特徴とするアクセス制御システム。

【請求項３】

請求項２記載のアクセス制御システムであって、

前記アクセス制御ポリシーに記すアクセス実行手段は、プログラムであり、該プログラムのパス名と、該プログラムの特徴値との組合せで特定することを特徴

とするアクセス制御システム。

【請求項４】

請求項３記載のアクセス制御システムにおいて、

さらに、ファイルアクセス要求内容を登録するアクセスログファイルを備え、

前記アクセス制御手段は、前記ファイルアクセス要求を前記ポリシーファイルの記述と照合し、正当性の判定結果を前記監視手段に送信するとともに、

前記アクセス要求が許可されるものであれば、当該アクセス実行手段の特徴値を、前記監視手段に送信し、

前記アクセス要求が前記アクセス制御ポリシーに違反するものであれば、該ファイルアクセス要求内容を、前記アクセスログファイル登録することを特徴とするアクセス制御システム。

【請求項５】

請求項４記載のアクセス制御システムにおいて、

さらにオープンファイルテーブルを備え、

前記アクセス制御手段により正当であると判定された前記ファイルアクセス要求のアクセスタイプがオープンアクセスである場合、

前記アクセス制御手段からレスポンス情報として取得した、アクセスタイプと、前記アクセス対象となるファイルと、前記アクセス要求発行元と、アクセス実行手段とを特定する情報を、前記オープンファイルテーブルに登録する手段と、

前記アクセス要求としてリードアクセスまたはライトアクセスが発行された場合は、

前記オープンファイルテーブルを検索し、前記アクセス要求の正当性を判定する手段を備える

ことを特徴とするアクセス制御システム。

【請求項６】

請求項５記載のアクセス制御システムであって、

前記監視手段は、さらに、

前記オープンファイルテーブルに登録されていないリードアクセス要求またはライトアクセス要求を検知した場合は、前記アクセス制御手段を介して当該アク

セス要求内容を前記アクセスログファイルに登録する手段を備える
ことを特徴とするアクセス制御システム。

【請求項７】

請求項６記載のアクセス制御システムであって、

前記監視手段は、さらに、

ファイルクローズ要求を検知した場合には前記オープンファイルテーブルから該当する情報を削除する手段を備える
ことを特徴とするアクセス制御システム。

【請求項８】

請求項７記載のアクセス制御システムであって、

前記監視手段は、さらに、

前記ファイルアクセス要求が正当であると判定された場合、前記前記監視手段は前記アクセス実行手段の特徴値を算出し、と前記アクセス制御手段から受信した前記特徴値とを比較する手段と、

一致した場合には前記アクセス要求を許可する手段と、

一致しない場合には前記ファイルアクセス要求を無効にすると共に、前記アクセス制御手段を介して当該ファイルアクセス内容を前記アクセスログファイルに登録する手段とを備える
ことを特徴とするアクセス制御システム。

【請求項９】

ファイルへの入出力手段と排他的に使用する第１の記憶手段と管理する第１のＯＳと、排他的に使用する第２の記憶手段を管理する第２のＯＳと、前記第１のＯＳと第２のＯＳとの間でデータ通信するための通信手段とを具備した情報処理システムにおいて、

前記第１のＯＳが管理する前記ファイル入出力手段へ発行されたファイルアクセス要求を監視する監視手段を前記第１のＯＳに設け、

前記ファイルアクセス要求の正当性をアクセス制御ポリシーに基づき判定するアクセス制御手段を前記第２のＯＳに設け、

前記監視手段は、前記アクセス要求を、前記通信手段を介して、前記アクセス

制御手段に伝え、正当性判定結果を、前記通信手段を介して、前記アクセス制御手段から受信することを特徴とする情報処理システム。

【請求項１０】

請求項９記載の情報処理システムにおいて、

前記第２のＯＳは、前記アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を記すポリシーファイルを備えることを特徴とする情報処理システム。

【請求項１１】

ファイルへの入出力手段と、前記ファイルを格納する記憶手段とを備えた情報処理装置に読み込まれ、実行され、前記情報処理装置上にアクセス制御システムを構成させるプログラムと前記プログラムが使用するファイルとを格納した記憶媒体であって、

前記プログラムが使用するファイルは、アクセス制御ポリシーを記述したポリシーファイルであって、

前記アクセス制御ポリシーは、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を記したものであり、

前記プログラムは、アクセス制御プログラムと、監視プログラムとを備え、

前記アクセス制御プログラムは、前記情報処理装置に、

当該ファイルアクセス要求の正当性を前記アクセス制御ポリシーに基づき判定させ、

前記監視手段は、前記情報処理装置に、

前記入出力手段を用いたファイルアクセス要求の発行を監視させ、

発行されたファイルアクセス要求を、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を用いて、前記アクセス制御プログラムに伝えさせること

を特徴とする記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、情報処理装置が管理する情報を不正なアクセスから保護する場合に好適なアクセス制御技術に関する。

【０００２】

【従来の技術】

一般のコンピュータシステムでは、マルチユーザー・マルチタスクＯＳが備えるユーザー認証機構と、該認証結果に基づいたファイルアクセス制御機構を用いて機密情報ファイルの保護を実現しているケースが多い。具体的には、前記ＯＳが実装された情報処理装置を利用する際に、ユーザーは必ず自己のユーザーＩＤとパスワードを入力し、認証を受ける。前記情報処理装置が管理する全てのファイル各々には、ファイルリードやライト等のアクセスタイプ毎に、ユーザーＩＤとグループＩＤを用いてアクセス可能なユーザーを定義したアクセスコントロールリストがセキュリティ属性情報として割り当てられている。。ユーザーがアプリケーションプログラムを介してファイルへアクセスした場合、上記ＯＳは、アクセス要求元となるユーザーのＩＤ及び該ユーザーが所属するグループＩＤを、アクセス対象となるファイルに割り当てられたアクセスコントロールリストと照合し、当該リストに前記ユーザーが含まれている場合に限りアクセスを許可するといった制御を行なう。

【０００３】

一方で、インターネットを介した情報発信、情報収集や、各種サービス提供を行なう手段としてＷｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ（WWW）のサービスが広く使われている。WWWでは、Ｈｙｐｅｒ　Ｔｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ（ＨＴＴＰ）と呼ばれる通信プロトコルが、リクエストデータとレスポンスデータの転送に用いられている。また、WWWシステムではコンテンツの不正な差し替えや、ネットワークを経由した機密情報流出を未然に防ぐためのセキュリティ技術がいくつか用意されている。

【０００４】

ＨＴＴＰが有するセキュリティ機構として、基本認証（Ｂａｓｉｃ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ）と呼ばれるものがある。基本認証では、認証情報としてユーザーＩＤとパスワードを予めWWWサーバに登録しておき、ブラウザを通じてユーザーから送信されるユーザーＩＤとパスワードを、前記認証情報と比較して認証を行なう。各コンテンツへのアクセス権限設定を記述したポリシーファイルと、該ポリシーに基づくアクセス制御機構も前記WWWサーバに実装されている。同様な機構を、Ｃｏｍｍｏｎ　Ｇａｔｅｗａｙ　Ｉｎｔｅｒｆａｃｅ（ＣＧＩ）プログラムに実装し、ユーザー認証とコンテンツへのアクセス制御を実現することも可能である。

【０００５】

WWWでは、他のセキュリティ技術として、認証局から発行される証明書に基づき、ユーザーとWWWサーバの相互認証と通信データの暗号化を行なうことが可能である。この技術は、消費者のクレジットカード番号がネットワーク上に流れる一部のインターネットショッピングのようなサービスにおいて、必須の技術とされている。これらの技術によるユーザー（クライアント）認証は、WWWサーバ側のＯＳが具備するユーザー認証機構とは通常独立したものであるが、前記認証に用いるユーザーＩＤや、ユーザーの証明書を、ＯＳが管理するユーザーアカウントに関連付けて使用することができるWWWサーバプログラムもある。つまり、認証されたユーザーは、ＯＳのアクセス制御下でWWWコンテンツにアクセスする。これらのセキュリティ技術については、“Web Security: A Step-by-Step Reference Guide” Lincoln D. Stein著、Addison-Wesley Pub Co; (ISBN: 0201634899) などに記載されている。

【０００６】

近年増加しているインターネットサービスプログラム等に潜むセキュリティホールやバグを利用した不正アクセスを解決する他の技術としては、常駐型のファイル監視プログラムを情報処理装置上に設け、定期的にファイルの破壊の有無をチェックする方法が、特開平１０－０６９４１７号公報に開示されている。これと同様の技術として、特定のファイルについてそのサイズの増減やタイムスタン

プを定期的にチェックするプログラムを情報処理装置上に設け、該ファイルの変化を検知するツールが、シェアウエアやフリーウエアとして幾つか公開されている。

【０００７】

【発明が解決しようとする課題】

先に述べたセキュリティホールやバグに関する情報は、ｈｔｔｐ：／／ｗｗｗ．ｃｅｒｔ．ｏｒｇ／　等で報告され、バグ修正用のプログラムも各メーカーから配布されるようになっている。しかし、攻撃者あるいは侵入者は、あらゆる手段を用いて外部ネットワークから情報処理装置への侵入を試みる。たとえば、前記マルチユーザー・マルチタスクなＯＳが具備するアクセス制御機構では、アクセス要求元のユーザーＩＤとその所属グループに基づいてアクセスの可否を判断しているため、ＯＳの管理者のような強い権限をもつユーザーに成りすまして侵入すれば、システム中のいかなるファイルにもアクセスできるので、システム中の情報を書き換えたり、機密情報を盗み出すことがてきてしまう。

【０００８】

また、前記ファイルの変化を定期的にチェックする手法では、ファイルが改ざんされた後の検出になるため、不正アクセスが発生したことは分かるが、それを未然に防ぐことができないと共に、ファイルの不正な読み出しについては事後検出もできない。

【０００９】

また、前記ＷＷＷサーバやＣＧＩプログラムを利用してコンテンツのアクセス制御機能を実現したシステムにおいては、各コンテンツへのアクセス権限設定を記述したポリシーファイルも、その他のファイルと同じく不正アクセスの対象になる。したがって、該ポリシーファイルが破壊あるいは改ざんされた場合には、前記アクセス制御機能が正常に働かない。

【００１０】

本発明の目的は、アクセス要求元（サブジェクト）がいかなる権限をもつ場合でも、そのアクセス手段およびアクセス対象（オブジェクト）を制限することが可能なアクセス制御システム及びその方法を提供することにある。

【００１１】

本発明の他の目的は、不正なファイルアクセスを未然に防ぐことが可能なアクセス制御システム及びその方法を提供することにある。

【００１２】

本発明の他の目的は、アクセス権限設定を記述したポリシーファイルと、該ポリシーファイルに基づいてアクセス制御を施行するプログラムを、外部からの不正アクセスや攻撃から保護可能なアクセス制御システム及びその方法を提供することにある。

【００１３】

【課題を解決するための手段】

上記目的を達成するために、本発明では、アクセス制御ポリシーとして、より厳密にアクセス要求を特定する情報、すなわち、アクセス要求元とアクセス実行手段とアクセスタイプとを特定する情報を用いる。

【００１４】

さらに具体的には、特定のファイルへのアクセスを、特定のユーザーが特定のプログラムを用いた場合のみ許可するアクセス制御ポリシーを記述したポリシーファイルを作り、アクセス制御手段が、ファイルへのアクセスが発生したときに、前記ポリシーファイルの記述に従ってアクセスの正当性すなわち可否を判定する。前記アクセス制御ポリシーには、アクセス対象となるファイルの名称と、アクセスが許可されたユーザー名とプログラム名の組合せを、ファイルのオープン・リード・ライト・削除・リネームといったアクセスタイプ毎に予め規定しておく。

【００１５】

また、不正なファイルアクセスを未然に防ぐために、本発明では、前記ファイルアクセスを監視するファイルＩ／Ｏ（Ｉｎｐｕｔ／Ｏｕｔｐｕｔ）フック手段を前記情報処理装置に設け、該ファイルＩ／Ｏフック手段は、前記アクセスを検知した時に、当該アクセスタイプ及びアクセス対象となるファイルの名称と、該アクセスの要求元となるユーザー名及びプログラム名を取得して前記アクセス制御手段に送信する。前記アクセス制御手段は、受信内容を前記ポリシーファイル

の内容と照合し、該ポリシーに違反するアクセスであれば当該アクセスを無効にし、前記ファイルＩ／Ｏフック手段を経由して前記アクセス要求元にエラーを返す。

【００１６】

また、前記ポリシーファイルと前記アクセス制御手段を保護するために、本発明では、１台の情報処理装置上に２つのＯＳと、両ＯＳ間でデータ通信するためのプロセス間通信手段と、両ＯＳが互いに排他的に占有するメモリや磁気ディスクおよびネットワークデバイスを設ける。前記２つのＯＳのうち一方をアクセス監視対象となるサービス用ＯＳとして使用し前記ファイルＩ／Ｏフック手段を前記サービス用ＯＳのカーネルレベルのモジュールとして設ける。もう一方をセキュリティ用ＯＳとして使用し、前記アクセス制御手段および前記ポリシーファイルを占有する磁気ディスクに格納すると共に、前記アクセス制御手段をプロセスとして動作させ、該アクセス制御手段とポリシーファイルを前記サービス用ＯＳ上の、サービスを受けるユーザが使うプロセスからアクセスできないようにする。

【００１７】

なお、本発明におけるＯＳとは、ユーザーまたはプログラムからの要求に応じて記憶媒体中のデータ、ファイルへのアクセスを実行する機能と、アクセス要求元のユーザーやプログラムを識別する機能と、排他的に占有する記憶手段を有するプログラムモジュールを意味しており、

・データ（ファイル）アクセスを管理しており、検知が可能である。

【００１８】

・アクセス要求元のユーザーを識別できる。

【００１９】

・アクセス要求元のアプリケーションを識別できる。

【００２０】

といった特徴を持つ。したがって、一般的にＯＳと呼ばれるものに限らず上記特徴を持つものであれば、本発明を適用することは可能である。

【００２１】

【発明の実施の形態】

以下、図を用いて本発明の実施の一形態を説明する。

図１は、本発明のアクセス制御システムの一構成例である。１００はサーバ情報処理装置であり、サービス用オペレーティングシステム（ＯＳ）１０３が管理するメモリ１０１、当該サービス用ＯＳ１０３が占有するディスクコントローラ１１２ａと磁気ディスク１１４ａとＬＡＮコントローラ１１３、セキュリティ用ＯＳ１０４が管理するメモリ１０２と、当該セキュリティ用ＯＳ１０４が占有するディスクコントローラ１１２ｂと磁気ディスク１１４ｂその他を備える。

【００２２】

サーバ情報処理装置１００の起動時には、サービス用ＯＳ１０３とサーバプログラム１０９がディスクコントローラ１１２ａを介して磁気ディスク１１４ａからメモリ１０１上にロードされると共に、セキュリティ用ＯＳ１０４とアクセス制御プログラム１１０がディスクコントローラ１１２ｂを介して磁気ディスク１１４ｂからメモリ１０２上にロードされる。前記サービス用ＯＳ１０３中のプロセス間通信プログラム１０８は、ファイルＩ／Ｏフックプログラムと前記アクセス制御プログラム１１０との間でデータ交換を行なうために使用するものとし、前記サーバプログラム１０９からは直接使用できないインタフェースとなっている。

【００２３】

また、サービス用ＯＳ１０３は、流通している既存のマルチユーザー・マルチタスクなＯＳと同様、ユーザーの識別・認証機能を具備しているものとする。前記サーバ情報処理装置１００と複数のクライアント情報処理装置１２０は、ローカルエリアネットワーク（ＬＡＮ）１１５を介して相互に接続されており、前記サーバプログラム１０９は、ＬＡＮコントローラ１１３とＬＡＮ１１５を介して、クライアントプログラム１２１からのリクエストの受信と、当該クライアントプログラム１２１へのレスポンスの送信を行なう。サーバプログラム１０９とクライアントプログラム１２１は、例えばＷＷＷサーバとブラウザに相当するプログラムである。なお、前記サーバ情報処理装置１００とクライアント情報処理装

置１２０は、電話回線やインターネットを介して接続されていてもよい。また、サービス用ＯＳとセキュリティＯＳとは、ＯＳ間通信のインタフェースを非公開にすれば別々の装置上にあっても良く、セキュリティ的にも問題はない。

【００２４】

前記サーバプログラム１０９のようにＯＳ上で動作するアプリケーションプログラムは、一般にユーザーレベルのプロセスと呼ばれている。該サーバプログラム１０９は、前述のようにクライアントプログラム１２１からのリクエストに基いて処理を実行するものであり、言い換えればネットワークからの攻撃の対象にもなり得るプログラムである。これに対し、ファイルＩ／Ｏフックプログラム１０６やファイルシステムドライバ１０７のようにＯＳの一機能として動作するプログラムは、一般にカーネルレベルのモジュールと呼ばれており、多くのコンピュータシステムではアクセス制御機能をカーネルレベルのモジュールとして実装している。

【００２５】

図１のサーバ情報処理装置１００のように、１台の情報処理装置上に複数のＯＳを同時に動作させるための技術としては、仮想計算機あるいはマイクロカーネルがある。その他、リアルタイムＯＳを一般的なＯＳのカーネルレベルのモジュールとして実装すると共に、前記リアルタイムＯＳがシステム障害を検知すると、リアルタイム処理を継続しながらもシステムを自動的に再起動する方法が、特開平１１－０２４９４３号公報に開示されている。また、プロセス間通信プログラム１０８のような、異種ＯＳ上のプロセス間通信を実現する方法は、特開平１１－０８５５４６号公報に開示されている。これらを本発明の前提条件として、引続き実施の形態を述べる。

【００２６】

図２は、前記ポリシーファイル２００のデータ構造を示したものである。

【００２７】

２１０から２１２は、ポリシー記述の一例を示したものである。オブジェクト名２０１は、アクセスを制限すべきファイルの名称を示す。オブジェクト名として、２１０や２１２のようにファイル単位の指定と、２１１のようなディレクトリ

単位の指定が可能となっている。ディレクトリ単位の指定では、該ディレクトリ名に続くファイル名を所定の文字、記号（たとえばアスタリスク（＊））で表記する。前記オブジェクトに対して禁止すべきアクセスを、禁止されたアクセスのタイプ２０２に示す。エラーコード２０３は、前記禁止されたアクセスのタイプ２０２が発生した際に、当該アクセス発行元（サブジェクト）となるプログラムに返すべきエラーコードを示す。例外サブジェクト２０４は、特別にアクセスを許されたプログラムの名称である。プログラムのハッシュ値２０５は、前記例外サブジェクトとして指定されたプログラムファイルの特徴値（ハッシュ値）をたとえば８バイトで表したものである。ユーザー名２０６は、前記例外サブジェクト２０４のプログラムを利用可能なユーザーを、前記サービス用ＯＳが管理するユーザー名またはグループ名で表したものである。

【００２８】

つまり前記ポリシーは、前記プログラムのハッシュ値２０５に登録されたハッシュ値を有するプログラムが前記例外サブジェクトとして登録され、且つ該プログラムをユーザー名２０６に登録されたユーザーあるいはグループのメンバーが利用している場合に限り、例外としてアクセスを許可することを表している。同時に、これら条件が整っていない場合は、アクセス発行元のプログラムに対して前記エラーコード２０３を返すことを意味する。

【００２９】

前記ポリシーファイル２００の設定は、システムのセキュリティ管理者が行なうものとする。例えば、ＨＴＭＬファイルなどのWWWコンテンツを侵入者によって不正に書き換えられないようにするには、該ＨＴＭＬファイルへのライトアクセスを原則として禁止しておき、例外としてコンテンツ管理者に任命されているユーザーが特定のＨＴＭＬ編集用プログラムを用いた場合に限りライトアクセスを許可するといったポリシーを前記ポリシーファイル２００に記述すればよい。

【００３０】

例外サブジェクトには、サーバプログラム１０９のように、ネットワークからの攻撃を受けやすいプログラムを指定しないことが重要である。更に例外サブジ

ェクトとして登録するプログラムを、ＣＤ－ＲＯＭ等の取り外し可能な記憶媒体に格納しておき、必要なときだけ媒体を装着して使用すればより確実に保護できるので、さらなるセキュリティ効果が期待できる。図１の構成に当該媒体の駆動装置を追加し、ディスクコントローラ１１２ａが当該駆動装置に対応すれば、このような記憶媒体が使用可能となる。

【００３１】

図３は、アクセスログファイル３００のデータ構造の一例を示したものである。アクセスログファイル３００は、前記ポリシーファイル２００にて禁じられているアクセスが発生した事実を、アクセス制御プログラム１１０のアクセスログ登録ルーチン４０６が書き込むためのファイルであり、当該アクセスが発生した日時３０１と、当該アクセスの対象となったファイルを表すオブジェクト名３０２と、当該アクセスのタイプ３０３、当該アクセスを発行したプログラムを表すサブジェクト名３０４、そして前記プログラムを利用していたユーザーを表すユーザー名３０５から構成される。

【００３２】

図４に、ファイルＩ／Ｏフックプログラム１０６と、アクセス制御プログラム１１０の構成を示す。

ファイルＩ／Ｏフックプログラム１０６は、前記サーバ情報処理装置１００の起動時に、サービス用ＯＳ１０３と共にメモリ１０１上にロードされる。アクセス制御プログラム１１０は、前記サーバ情報処理装置１００の起動時に、セキュリティ用ＯＳ１０４と共にメモリ１０２上にロードされる。

【００３３】

図５と図６を用いて、本発明のアクセス制御の概要を記す。

【００３４】

図５は、Ｉ／Ｏマネージャ１０５と、ファイルＩ／Ｏフックプログラム１０６と、アクセス制御プログラム１１０の三者間における処理の流れを示すものである。５０１は、サーバプログラム１０９が発行したファイルアクセスが、Ｉ／Ｏマネージャ１０５を経由してファイルＩ／Ｏフックプログラム１０６に到達するまでの通信経路であり、サービス用ＯＳ１０３の上で動作する全てのアプリケー

ションプログラムからのファイルアクセスについて共通なものである。ファイルＩ／Ｏフックプログラム１０６には、図６に示すＩ／Ｏパケット６００のデータが渡される。Ｉ／Ｏパケット６００は、ファイルＩ／Ｏフックプログラム１０６に渡された時点では、既にサービス用ＯＳが具備するアクセス制御機構はパスしている。つまり、サービス用ＯＳが具備するアクセス制御によって許可されたアクセスに関するパケットである。

【００３５】

図６は、前記三者間を流れるパケットのデータ構造を示したものである。６０１はアクセス対象のファイルを表すオブジェクト名である。６０２は前記オブジェクトへのアクセスタイプを示す。６０３は、当該アクセスを発行したプログラムのプロセスＩＤを示す。６０４は当該アクセスの処理結果を示すステータスを示す。６０５は当該アクセスに関連するデータの長さを示すものであり、例えば前記アクセスタイプ６０２がファイルへの書込み（ライト）要求であれば、書込みデータがデータ領域６０７に格納され、当該書込みデータの長さがデータ長６０５に格納される。６０６は、前記データ領域６０７へのポインタである。

【００３６】

図５において、５０２は、ファイルＩ／Ｏフックプログラム１０６が検知したファイルオープン要求に関するアクセス権限チェックと、アクセスログ３００への書き込みを、アクセス制御プログラム１１０に対してリクエストするに用いる通信経路である。前記リクエストのデータ構造を、図６のリクエストパケット６１０に示す。サブジェクト名６１１は、ファイルアクセスの発行元となるプログラムの名称であり、ユーザー名６１２は、前記プログラムの利用者をユーザー名とグループ名で表すものである。サブジェクト名６１１及びユーザー名６１２は、受信したＩ／Ｏパケット６００に含まれるプロセスＩＤ６０３を指定して、前記サービス用ＯＳとして用いる標準のＯＳに対してシステムコールを発行することで取得できる。

【００３７】

前記リクエストパケット６１０は、プロセス間通信プログラム１０８が当該パケットのデータを、ファイルＩ／Ｏフックプログラムのメモリ空間６３０からア

クセス制御プログラムのメモリ空間６４０に複写することで伝達される。アクセス制御プログラム１１０では、受信したリクエストパケットに含まれる情報を、ポリシーファイル２００の内容と照合し、その結果をレスポンスパケット６２０に登録して前記ファイルＩ／Ｏフックプログラムに伝達する。この伝達は、前記通信経路５０２と同様に、プロセス間通信プログラム１０８が前記パケットのデータを、アクセス制御プログラムのメモリ空間６４０からファイルＩ／Ｏフックプログラムのメモリ空間６３０に複写することで達成される。５０３は、当該レスポンスパケットの通信経路を示すものである。

【００３８】

ファイルＩ／Ｏフックプログラム１０６は、受信したレスポンスパケット６２０の内容から当該アクセスの可否を判断し、アクセス不可であればエラーコード２０３をＩ／Ｏパケット６００のステータス６０４に設定し、当該Ｉ／Ｏパケットを前記Ｉ／Ｏマネージャ１０５に返す。前記ステータス６０４にエラーコードが設定されている場合、前記Ｉ／Ｏマネージャ１０５は、通信経路５０６を用いてファイルアクセス発行元のサーバプログラム１０９に当該エラーを返す。一方、前記ステータス６０４にエラーコードが設定されていなければ、通信経路５０５を用いてファイルシステムドライバ１０７及びディスクコントローラ１１２ａを介して磁気ディスク１１４ａへのファイルアクセスを続行し、通信経路５０６を用いて当該アクセスの結果を前記サーバプログラム１０９に返す。

【００３９】

本発明のアクセス制御の具体的処理内容を、図７から図１６を用いて説明する。

図７は、ファイルＩ／Ｏフックプログラム１０６が具備するファイルＩ／Ｏフックルーチンの処理内容を示すフロー図である。

サーバプログラム１０９が、ステップ７０１でクライアントプログラム１２１からのリクエストを受信し、ステップ７０２で前記リクエストに応じたファイルアクセスをサービス用ＯＳ１０３に対して発行した場合を想定する。

【００４０】

当該ファイルアクセスは、通信経路５０１を経由して、ファイルＩ／Ｏフック

ルーチン４００にＩ／Ｏパケット６００として渡される（ステップ７０３）。ステップ７０３からステップ７１２は、ファイルＩ／Ｏフックルーチン４００の処理フローである。ステップ７０４では、Ｉ／Ｏパケット６００に含まれるプロセスＩＤ６０３から、ファイルアクセス発行元のサブジェクト名６１１とユーザー名６１２を取得する。

【００４１】

ステップ７０５からステップ７０８では、アクセスタイプ６０２を調べ、ファイルオープン、ファイルクローズ、ファイルリード又はライト、またはファイル削除又はリネームにそれぞれ対応する処理ルーチンを実行する。これらに該当しない場合、又は対応する処理ルーチンを実行した後は、通信経路５０４を経由して、ステップ７１３にてＩ／Ｏマネージャ１０５の処理に戻る。

【００４２】

図８を用いてオープン処理ルーチン４０１の処理フローを説明する。ステップ８０１では、アクセス日時として現在の日付と時刻を取得する。ステップ８０２では、リクエストパケット６１０を作成すると共に、当該パケット中のエラーコード２０３とハッシュ値２０５を０に初期設定する。次にステップ８０３にて、アクセス制御プログラム１１０のオープン制御ルーチン４０５をコールすると共に、前記リクエストパケット６１０をオープン制御ルーチン４０５に渡す。

【００４３】

オープン制御ルーチン４０５の処理フローを、図９を用いて説明する。ステップ９０１では、受信したリクエストパケット６１０のうち、オブジェクト名６０１とアクセスタイプ６０２の内容を、ポリシーファイル２００と照合する。ステップ９０２では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。このとき、禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、ステップ９０３にて、許可されたアクセスタイプをレスポンスパケット６２０の認可アクセス６１３に登録し、オープン処理ルーチン４０１の処理に戻る。ここで、許可されたアクセスタイプとは、オブジェクト名６０１で示されるファイルに対するリードとライトの内、サブジェクト名６１１で示されるプログラムから実行可能なアクセスタイプのことを表す。したがっ

て、オブジェクト名６０１で示されるファイルに対して、ポリシーファイル２００の中でリードとライト共に許可されていれば、認可アクセス６１３にはリードとライトの両方を登録する。また、リードのみが許可されている場合はリードのみを、ライトのみが許可されている場合はライトのみを前記認可アクセス６１３に登録する。リードとライトの両方を禁じる場合には、ポリシーとして当該ファイルのオープンを禁止するよう、予めポリシーファイル２００に記述しておく必要がある。

【００４４】

ステップ９０２にて禁止されたアクセスだと判断した場合、ステップ９０４にて、前記サブジェクト名６１１とユーザー名６１２が、共にポリシーファイル２００に記述された例外サブジェクト２０４とユーザー名２０６に該当するか否かを判別する。このとき、サブジェクト名については該プログラムファイルのパス名が完全に一致するか否かを判別する。またユーザー名については、ユーザー名６１２に含まれるユーザー名とグループ名のいずれかが一致するか否かを判別する。判別の結果、例外サブジェクト２０４かつユーザー名２０６に該当する場合、ステップ９０５にて当該プログラムファイルのハッシュ値２０５をポリシーファイル２００から取得して、レスポンスパケット６２０に設定する。また、ステップ９０６では、ステップ９０３と同様に、許可されたアクセスタイプをレスポンスパケット６２０の認可アクセス６１３として設定する。その後、ステップ９０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、オープン処理ルーチン４０１へ戻る。ここでのエラーコード設定は、オープン処理ルーチン４０１のステップ８０４からステップ８１０の処理の中で意味を持つものである。

【００４５】

ステップ９０４にて、例外サブジェクト２０４とユーザー名２０６とに該当しないと判断した場合、ステップ９０７にて、当該アクセスの内容をアクセスログファイル３００に書込む。その後、ステップ９０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、オープン処理ルーチン４０１へ戻る。

【００４６】

次に、図８において、ステップ８０４ではレスポンスパケット中のハッシュ値２０５の値を調べ、０であれば例外として認められたアクセスではないと判断し、ステップ８０５でエラーコードの値をチェックする。エラーコードが０でなければ禁止されたアクセスであったと判断し、ステップ８０６にて、Ｉ／Ｏパケット６００のステータス６０４に、当該エラーコード２０３を設定して処理を終了する。ステップ８０５で、エラーコードが０であれば、正当なアクセスであったと判断し、ステップ８１０にて当該アクセス情報を後述するオープンファイルテーブルの先頭アドレスに登録してから処理を終了する。

【００４７】

ステップ８０４でハッシュ値が０でない場合は、例外として認められたサブジェクトであると判断し、ステップ８０７にてサブジェクト名６０１で示されるプログラムファイルのハッシュ値を算出し、ステップ８０８にてレスポンスパケット中のハッシュ値２０５と比較する。両者が等しければ、例外サブジェクト２０４に相当するとみなして、ステップ８１２にて当該アクセス情報をオープンファイルテーブルの先頭アドレスに登録してから本ルーチンの処理を終了する。ハッシュ値が等しくなければ、前記プログラムファイルが不正なプログラムであるとみなして、ステップ８０９にて前記リクエストパケットにおけるサブジェクト名６１１を例外サブジェクトにならないように、例えばサブジェクト名としてnullデータに、変更し、、ステップ８１０にてアクセスログ登録ルーチン４０６をコールすると共に前記リクエストパケット６１０をアクセスログ登録ルーチン４０６に渡す。

【００４８】

該アクセスログ登録ルーチン４０６の処理フローを、図１４を用いて説明する。ステップ１４０１で、リクエストパケット６１０の内容をポリシーファイル２００と照合する。予め前記ステップ８０９で例外サブジェクト扱いにならぬようにサブジェクト名６１１をクリアしておくことにより、必ず禁止されたオープンアクセスとして扱われる。ステップ１４０２にて、該当するエラーコードをポリシーファイル２００から読み出してレスポンスパケット６２０に設定し、当該パ

ケットを返す。ステップ１４０３にて当該アクセス内容を、アクセスログファイル３００に書き込み、オープン処理ルーチン４０１の処理に戻る。

【００４９】

図８のオープン処理ルーチン４０１において、ステップ８１１でレスポンスパケット中のエラーコード２０３をＩ／Ｏパケットのステータス６０４に設定して本ルーチンの処理を終了する。

【００５０】

図１０を用いてオープンファイルテーブルについて説明する。オープンファイルテーブル１０００は、現在オープン中のファイルに関する情報を格納した構造体データ１００２の集合である。１個の構造体には１件のオープンファイルの情報を記憶しており、ファイルＩ／Ｏフックプログラム１０６では、各構造体を先頭アドレス１００１とポインタ１００３によりリストとして管理する。本オープンファイルテーブル１０００の一例を図１１に示す。

【００５１】

オープンファイルテーブル１０００に登録された情報に該当するアクセスであれば、該アクセスを許可することになるため、オープンファイルテーブル１０００を不正に書換えられないよう保護することが重要である。サービス用ＯＳ１０３として、プロセス毎に独立したメモリ空間が割当てられると共にメモリ空間の排他制御機構が働くＯＳを使うことで、前記オープンファイルテーブル１０００も別プロセスからは不正に書換えできない仕組みにすることが可能になる。

【００５２】

図１２は、クローズ処理ルーチン４０２の処理フローを示したものである。ステップ１２０１では、受信したＩ／Ｏパケット６００の中にあるオブジェクト名６０１と、プロセスＩＤ６０３、並びにプロセスＩＤ６０３から取得したサブジェクト名６１１およびユーザー名６１２の組み合わせと同じものをオープンファイルテーブル１０００から検索する。ステップ１２０２で、該当する情報がテーブルにあれば、ステップ１２０３にて該当する情報をテーブルから削除する。一方、テーブルに存在しなければ、ポリシーファイル２００に登録されていないファイルへのクローズ要求とみなし、本ルーチンの処理を終了する。

【００５３】

図１３は、リード・ライト処理ルーチン４０３の処理フローを表したものである。ステップ１３０１では、受信したＩ／Ｏパケット６００の中にあるオブジェクト名６０１と、プロセスＩＤ６０３と、プロセスＩＤ６０３から取得したサブジェクト名６１１と、ユーザー名６１２との組み合わせと同じものをオープンファイルテーブル１０００から検索し、更にアクセスタイプ６０２が認可されたアクセスタイプ６１３に含まれるか否かをチェックする。ステップ１３０２にて、前記オブジェクト名６０１と、プロセスＩＤ６０３と、サブジェクト名６１１と、ユーザー名６１２と、アクセスタイプ６０２との組合せがテーブル１０００にあれば、本ルーチンの処理を終了する。このことは、オープンファイルテーブル１０００に登録されたアクセスであれば正当なアクセスであるとみなし、アクセス制御プログラム１１０の処理を行わないことを意味する。これは、本発明のアクセス制御によって生じるシステムのパフォーマンス低下を軽減する効果がある。

【００５４】

ステップ１３０２にて、オープンファイルテーブルに登録されたアクセスでなければ前記ポリシーファイル２００により禁止されたアクセスとみなし、ステップ１３０３でアクセス日時を取得し、ステップ１３０４でリクエストパケット６１０を作成する。このとき、エラーコード２０３を０に初期設定する。この後、ステップ１３０５にてアクセスログ登録ルーチン４０６の処理にジャンプする。

【００５５】

図１４のアクセスログ登録ルーチン４０６において、ステップ１４０１ではリクエストパケット６１０の内容をポリシーファイル２００と照合し、ステップ１４０２にて、該当するエラーコードをポリシーファイル２００から読み出してレスポンスパケット６２０に設定する。次に、ステップ１４０３にて当該アクセス内容を、アクセスログファイル３００に書き込み、リード・ライト処理ルーチン４０３の処理に戻る。リード・ライト処理ルーチン４０３では、図１３のステップ１３０６において、レスポンスパケットで受信したエラーコードをＩ／Ｏパケットのステータス６０４に設定し、本ルーチンの処理を終了する。

【００５６】

図１５は、削除・リネーム処理ルーチン４０４の処理フローを表したものである。図８のオープン処理ルーチン４０１と同じ処理内容のステップには同じ番号を付している。

【００５７】

ステップ１５０３にて、アクセス制御プログラム１１０の削除・リネーム制御ルーチン４０７をコールすると共に、前記リクエストパケット６１０を削除・リネーム制御ルーチンに渡す。

【００５８】

削除・リネーム制御ルーチン４０７の処理フローを、図１６を用いて説明する。図９のオープン制御ルーチン４０５と同じ処理内容のステップには同じ番号を付している。ステップ１６０２では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、削除・リネーム処理ルーチン４０４の処理に戻る。

【００５９】

ステップ１６０２にて禁止されたアクセスだと判断した場合、図９と同様にステップ９０４、９０５の処理を行い、ステップ１６０３にて、例外サブジェクト２０４とユーザー名２０６に該当しないと判断した場合、図９と同様にステップ９０７の処理を行う。

【００６０】

ステップ９０５または９０７の処理を済ませると、ステップ９０８で、該当するエラーコード２０３をレスポンスパケット６２０に設定してから、削除・リネーム処理ルーチン４０４へ戻る。

【００６１】

次に図１５において、図８と同様にステップ８０４の処理を行う。ステップ１５０５では、エラーコードが０であれば、正当なアクセスであったと判断し、本ルーチンの処理を終了する。ステップ８０７にてサブジェクト名６０１で示されるプログラムファイルのハッシュ値を算出し、ステップ８０８にてレスポンスパ

ケット中のハッシュ値２０５と比較して、両者が等しければ、例外サブジェクト２０４に相当するとみなして本ルーチンの処理を終了する。ハッシュ値が等しくない場合は図８と同様の処理を行い、アクセスログ登録ルーチン４０６を呼び出す。当該ルーチンが処理を済ませると削除・リネーム処理ルーチン４０４の処理に戻る。

【００６２】

図１５において、ステップ１５１１でレスポンスパケット中のエラーコード２０３をＩ／Ｏパケットのステータス６０４に設定して本ルーチンの処理を終了する。

【００６３】

以上説明したように、本発明によれば、進入者が強い権限をもつユーザーに成りすました場合でも、ファイルへのアクセスを制限できるという効果がある。

また、不正なファイルアクセスを未然に防ぐことができるという効果がある。

また、前記ポリシーファイルとアクセス制御手段を、外部からの不正アクセスや攻撃から保護できるという効果がある。

また、図１に示すシステムの、サーバ情報処理装置とクライアント情報処理装置とを結ぶLAN１１５などの通信回線上に、ファイアウォールを設け、さらに本発明を併用すれば、より強固なセキュリティを確保することが出来る。

【００６４】

【発明の効果】

本発明によれば、情報処理装置が管理するファイル、情報を不正なアクセスから保護することが可能になる。

【図面の簡単な説明】

【図１】

本発明の実施の形態におけるアクセス制御システムの一構成例を示す図。

【図２】

アクセス制御ポリシーの設定を格納するためのポリシーファイルを示す図。

【図３】

不正アクセスが発生した事実を記録するためのアクセスログファイルを示す図

【図４】

ファイルＩ／Ｏフックプログラム１０６とアクセス制御プログラム１１０を構成するプログラムルーチンを示す図。

【図５】

本発明の実施の一形態において、各モジュール間を流れるデータの通信経路を示す図。

【図６】

本発明の実施の一形態において、各モジュール間を流れるデータの構造を示す図。

【図７】

ファイルＩ／Ｏフックルーチン４００の処理のフローチャートを示す図。

【図８】

オープン処理ルーチン４０１の処理のフローチャートを示す図。

【図９】

オープン制御ルーチン４０５の処理のフローチャートを示す図。

【図１０】

オープンファイルテーブルの構造体とそのリストを示す図。

【図１１】

オープンファイルテーブルに登録されるデータの一例を示す図。

【図１２】

クローズ処理ルーチン４０２の処理のフローチャートを示す図。

【図１３】

リード・ライト処理ルーチン４０３の処理のフローチャートを示す図。

【図１４】

アクセスログ登録ルーチン４０６の処理のフローチャートを示す図。

【図１５】

削除・リネーム処理ルーチン４０４の処理のフローチャートを示す図。

【図１６】

削除・リネーム処理ルーチン４０７の処理のフローチャートを示す図。

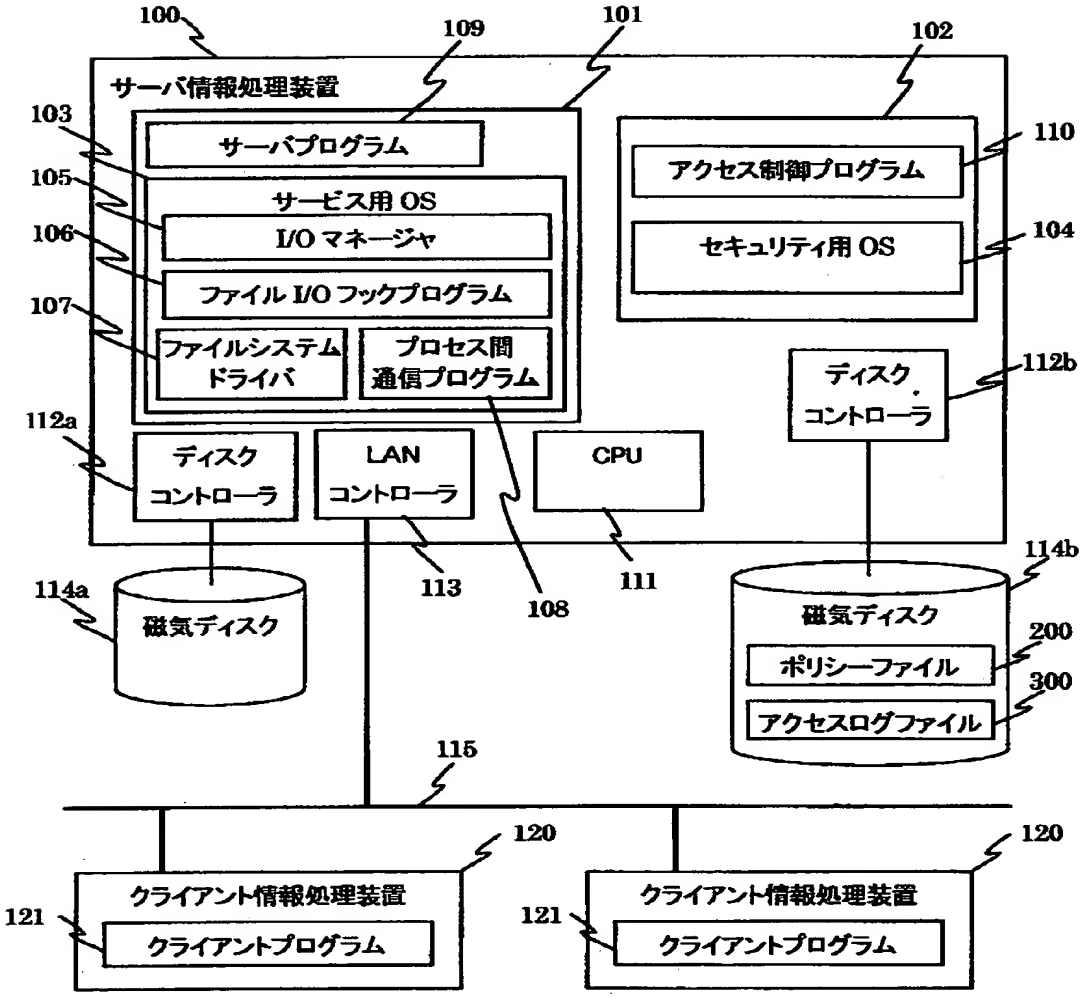
【符号の説明】

１００･･･サーバ情報処理装置、１０１･･･サービス用ＯＳが管理するメモリ、１０２･･･セキュリティ用ＯＳが管理するメモリ、１０３･･･サービス用ＯＳ、１０４･･･セキュリティ用ＯＳ、１０５･･･Ｉ／Ｏマネージャ、１０６･･･ファイルＩ／Ｏフックプログラム、１０７･･･ファイルシステムドライバ、１０８･･･プロセス間通信プログラム、１０９･･･サーバプログラム、１１０･･･アクセス制御プログラム、１１１･･･ＣＰＵ、１１２･･･ディスクコントローラ、１１３･･･ＬＡＮコントローラ、１１４･･･磁気デバイス、１１５･･･ＬＡＮ、１２０･･･クライアント情報処理装置、１２１･･･クライアントプログラム、２００･･･ポリシーファイル、３００･･･アクセスログファイル、５０１～５０６･･･通信経路、６００･･･Ｉ／Ｏパケット、６１０･･･リクエストパケット、６２０･･･レスポンスパケット、１０００･･･オープンファイルテーブル

【書類名】　図面

【図１】

図１



100：サーバ情報処理装置

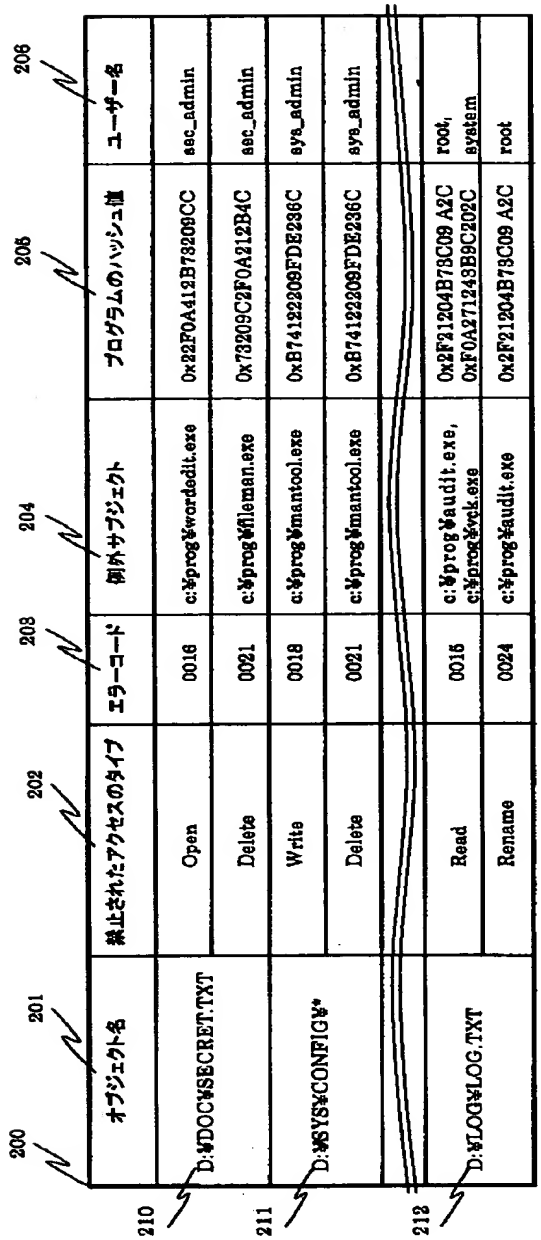
101：サービス用 OS が管理するメモリ

102：セキュリティ用 OS が管理するメモリ

120：クライアント情報処理装置

【図２】

図２



200: ポリシーファイル

【図３】

図３

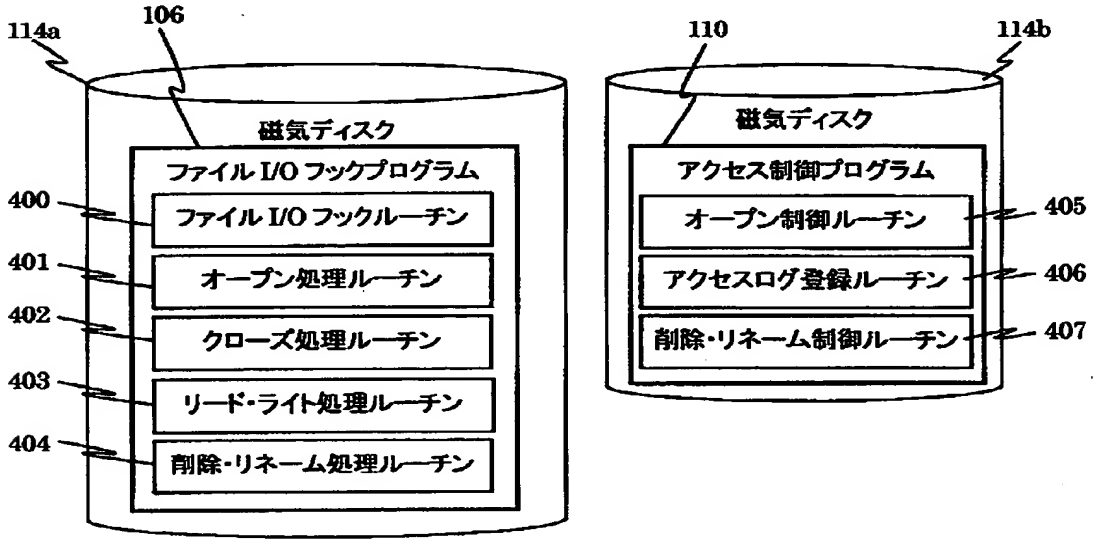
300

日時 (301)	オブジェクト名 (302)	アクセスタイプ (303)	サブジェクト名 (304)	ユーザー名 (305)
1999.07.28 15:32:46	D:¥DOC¥SECRET.TXT	Write	c:¥prog¥wwwsrv.exe	u_0023
1999.07.27 09:15:10	D:¥DOC¥SECRET.TXT	Delete	c:¥prog¥wwwsrv.exe	u_0023
1999.07.25 16:02:55	D:¥SYS¥CONFIG¥	Write	c:¥prog¥txtedit.exe	intruder
1999.07.25 14:44:28	D:¥SYS¥CONFIG¥	Delete	c:¥prog¥fileman.exe	intruder
1999.07.25 14:42:59	D:¥LOG¥LOG.TXT	Read	c:¥prog¥txtedit.exe	user007
1999.07.16 10:29:31	D:¥LOG¥LOG.TXT	Delete	c:¥prog¥fileman.exe	user007

300: アクセスログファイル

【図４】

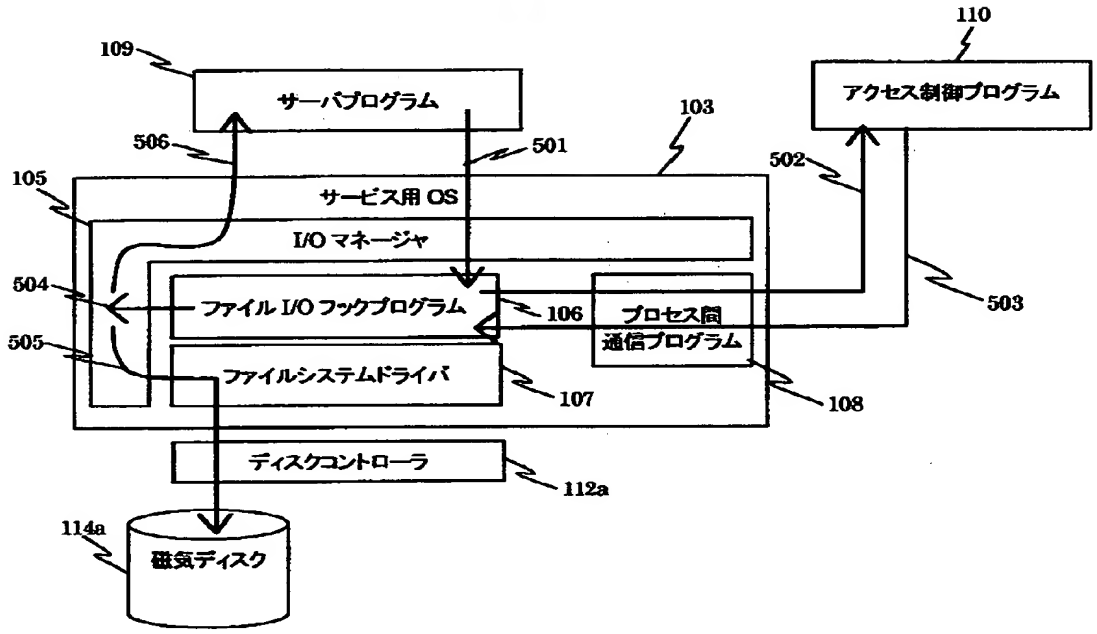
図４



114a, 114b: 磁気ディスク

【図５】

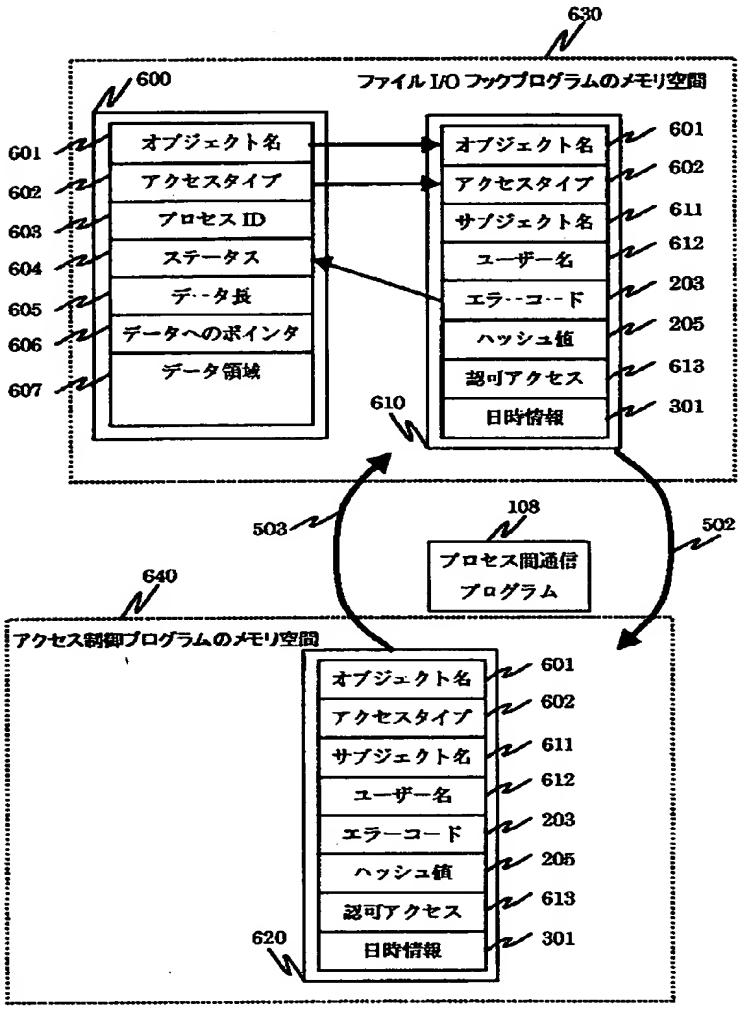
図５



501～506: 通信経路

【図６】

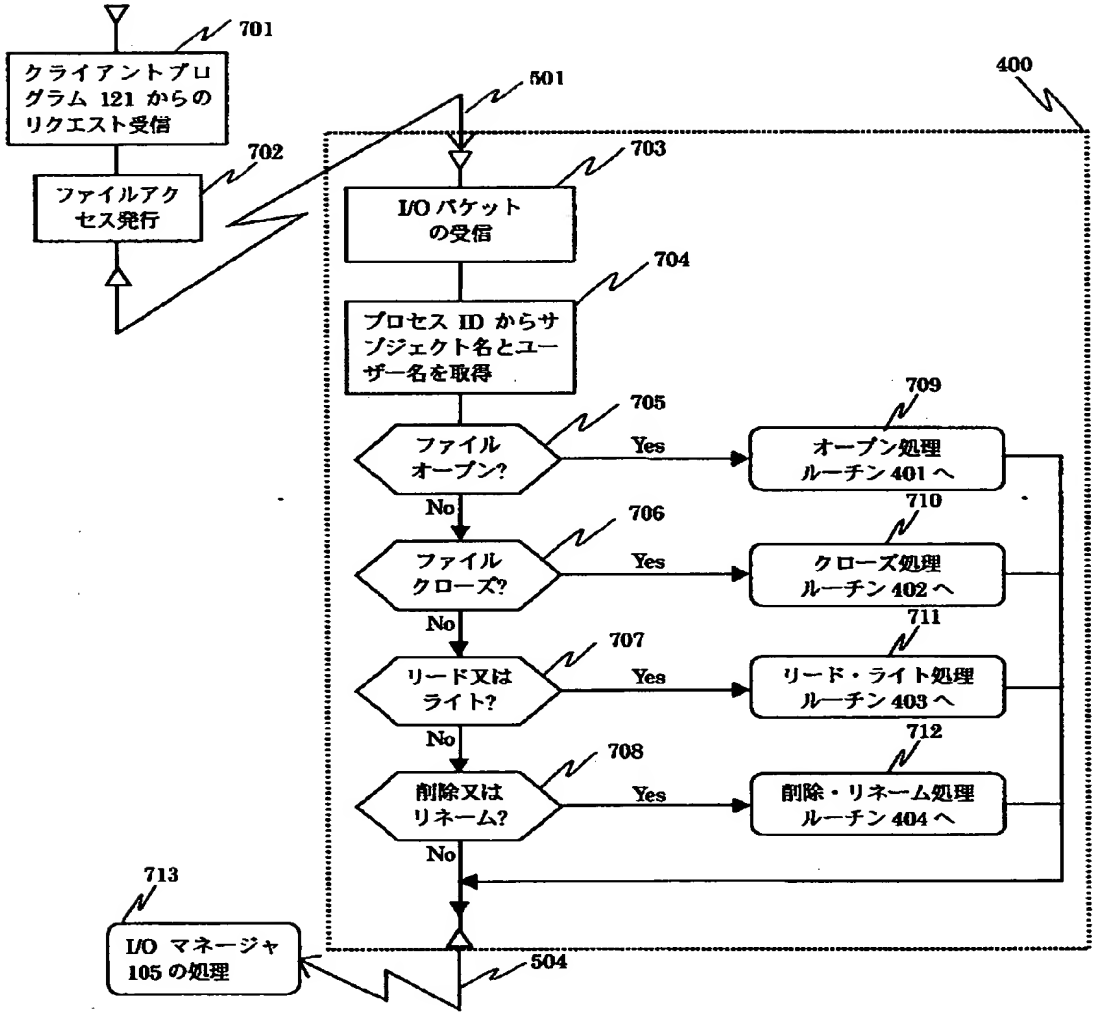
図 6



600: I/O パケット
610: リクエストパケット
620: レスポンスパケット
630: ファイル I/O フックプログラムのメモリ空間
640: アクセス制御プログラムのメモリ空間

【図７】

図７

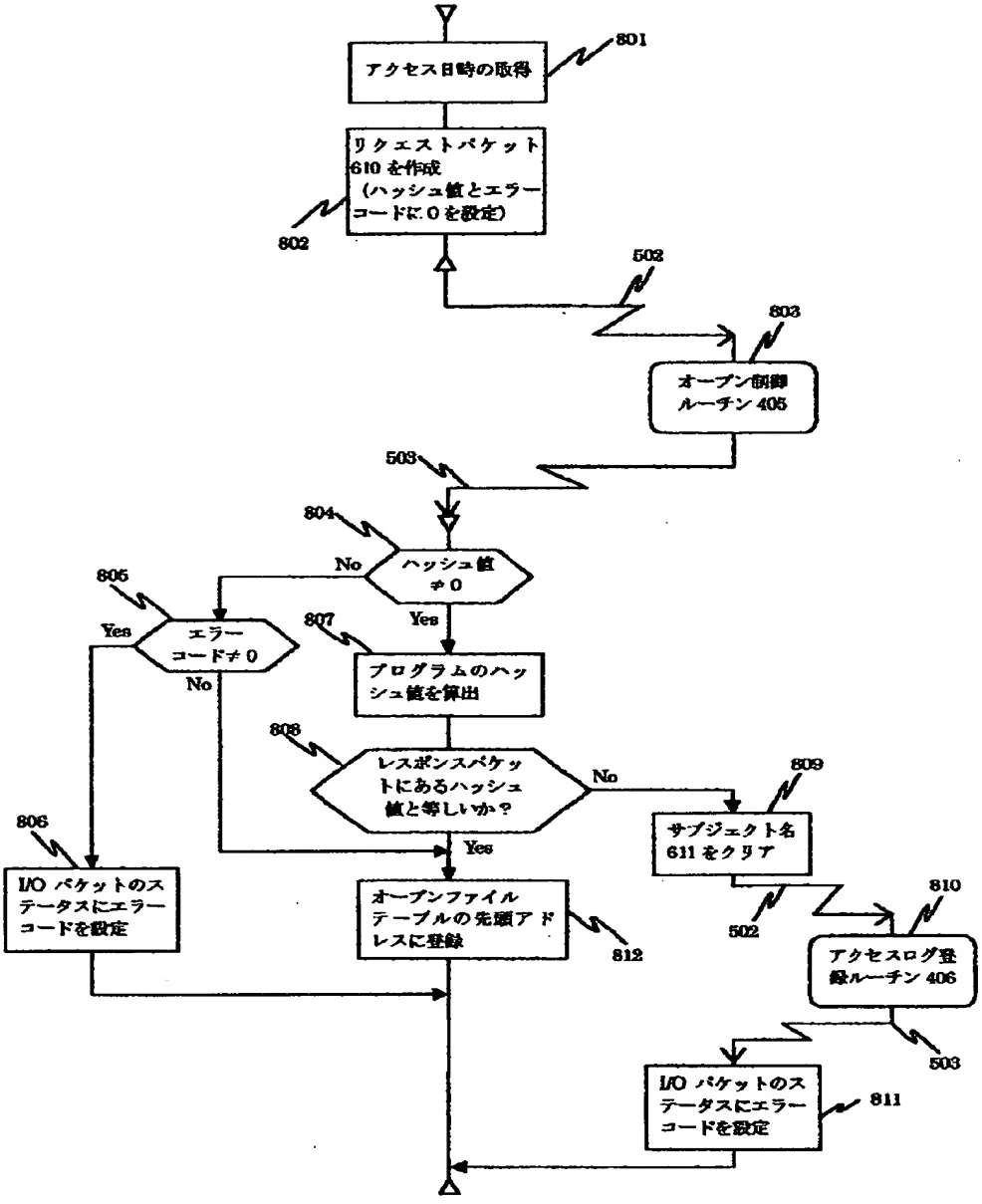


701～702: サーバプログラム 109 の概略処理フロー

703～712: ファイル I/O フックルーチン 400 の処理フロー

【図８】

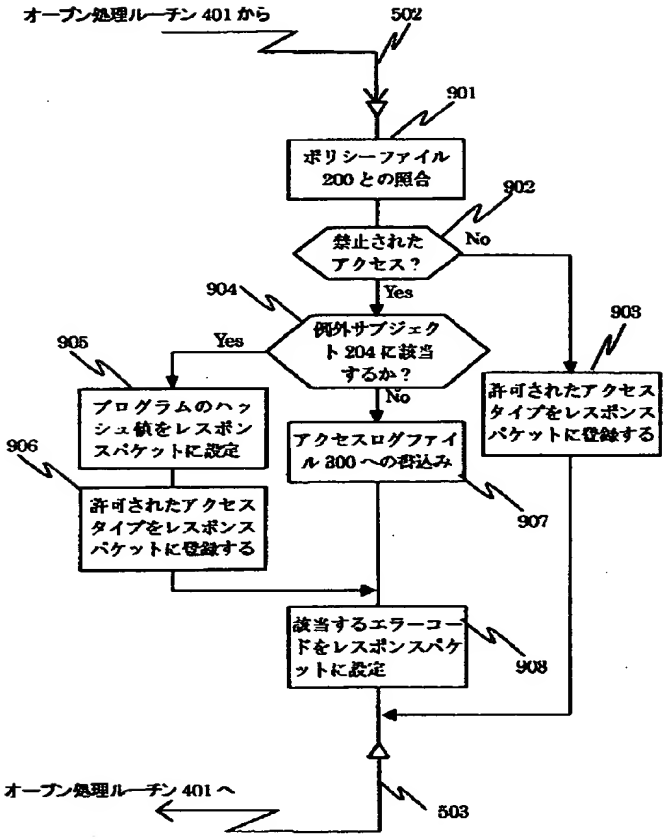
図８



801～810: オープン処理ルーチンの処理フロー

【図９】

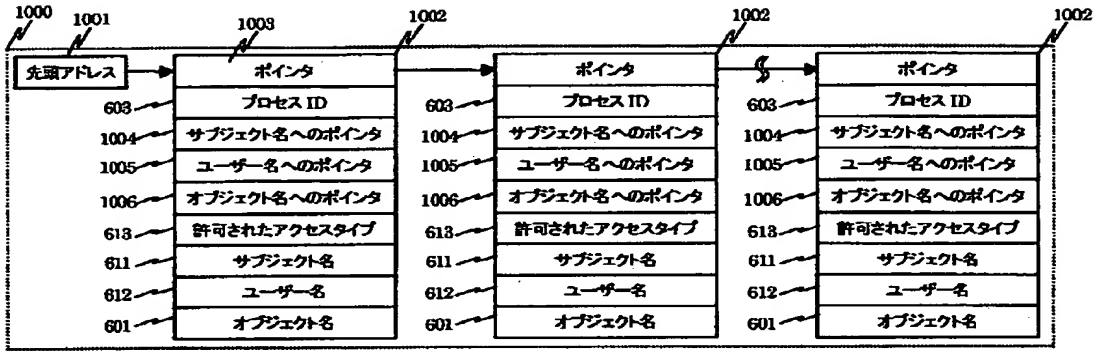
図 9



901～908: オープン制御ルーチン 405 の処理フロー

【図１０】

図１０



1000: オープンファイルテーブル
1002: オープンファイルの構造体データ

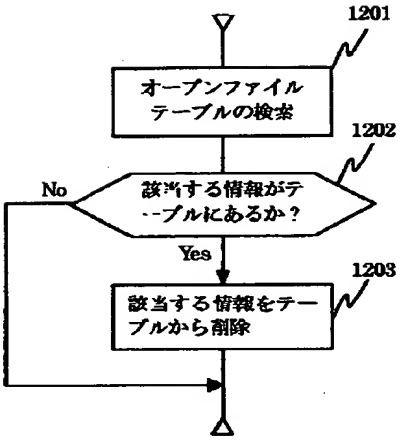
【図１１】

図１１

1000 / プロセス ID (603)	サブジェクト名 (611)	ユーザー名 (612)	オブジェクト名 (601)	認可されたアクセスタイプ (613)
0022	c:¥prog¥wwwserv.exe	inet	D:¥HOME¥SALES.HTML	Read
0043	c:¥prog¥wordedit.exe	sec_admin	D:¥DOC¥SECRET.TXT	Read/Write
0067	c:¥prog¥viewer.exe	tarou	D:¥DOC¥IDEA.TXT	Read
0092	c:¥prog¥mantool.exe	sys_admin	D:¥SYS¥CONFIG¥PORT	Read/Write
0113	c:¥prog¥vck.exe	system	D:¥DOC¥SHEET.DOC	Read
0218	c:¥prog¥audit.exe	root	D:¥LOG¥LOG.TXT	Read/Write

【図１２】

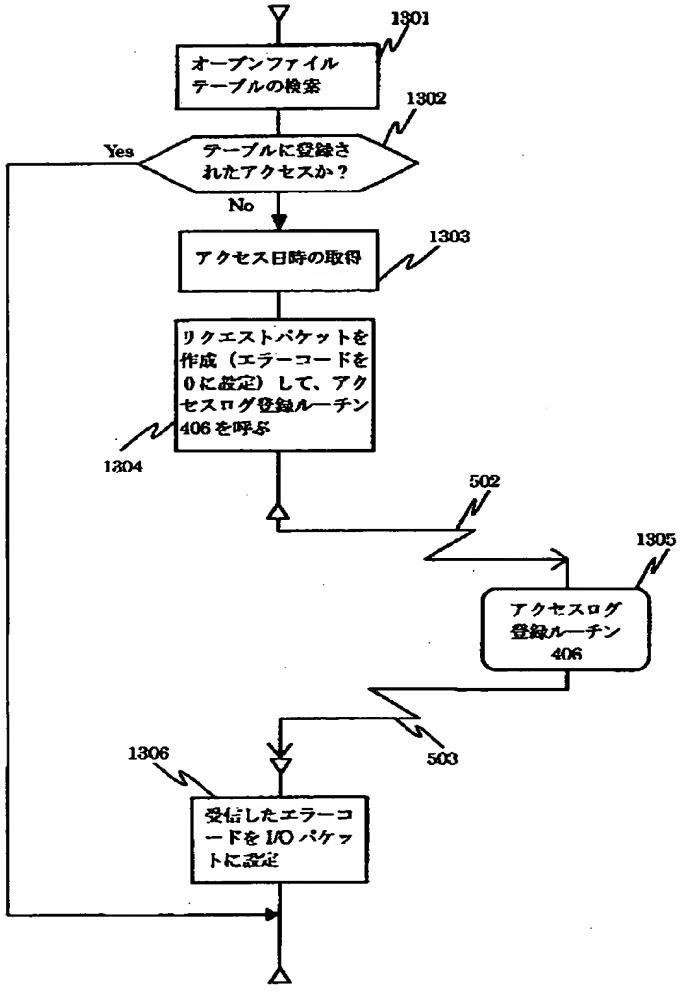
図１２



1201～1203: クローズ処理ルーチン 402 の処理フロー

【図１３】

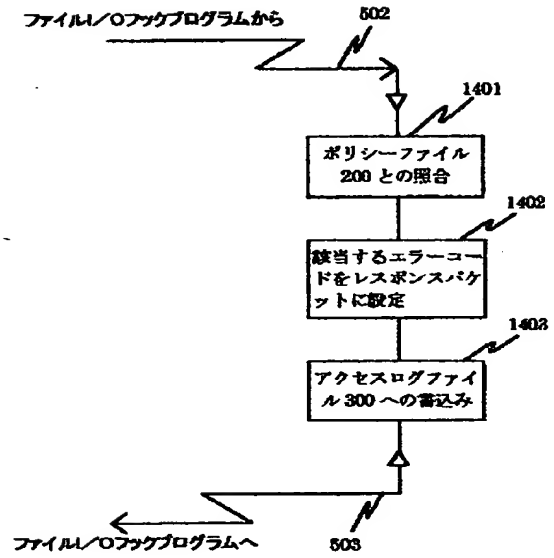
図１３



1301～1306: リード・ライト処理ルーチン 403 の処理フロー

【図１４】

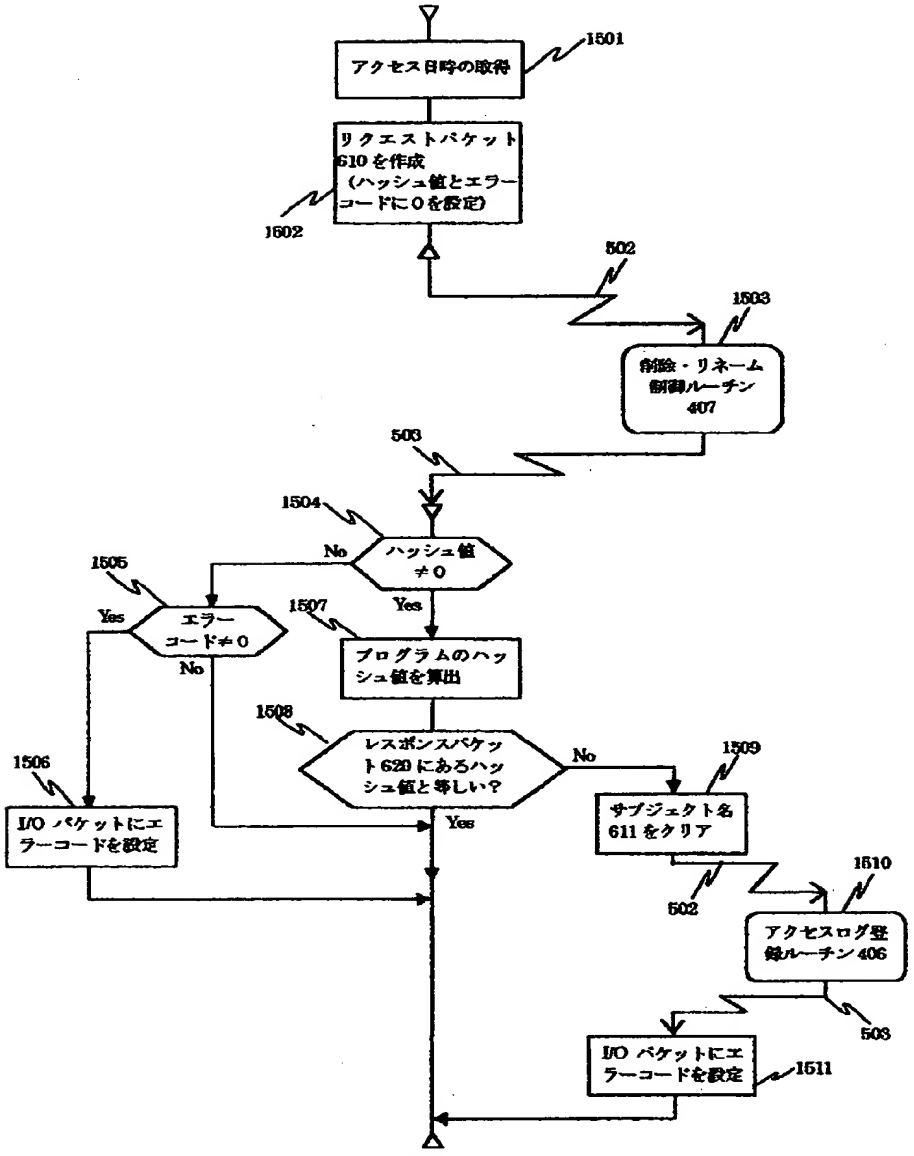
図１４



1401～1403: アクセスログ登録ルーチン406の処理フロー

【図１５】

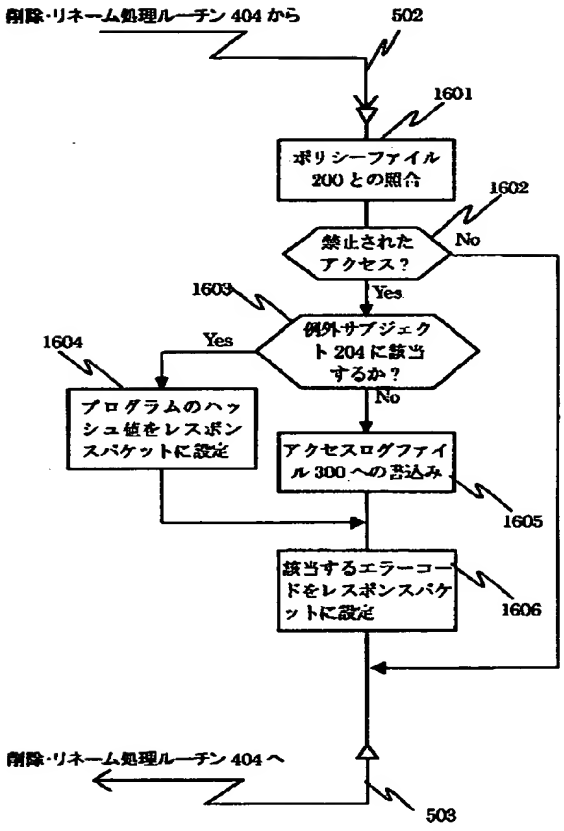
図１５



1501～1509: 削除・リネーム処理ルーチン 404 の処理フロー

【図１６】

図１６



1601～1606: 削除･リネーム制御ルーチン 407 の処理シーケンス

【書類名】　要約書

【要約】

【課題】

ネットワークからの侵入者がいかなるユーザー権限を利用して不正なファイル読み出しや書き込みを試みても、該アクセスの抑止が可能なアクセス制御システム及びその方法を提供する。

【解決手段】

特定のファイルへのアクセスを、特定のユーザーが特定のプログラムを使用した場合に限り許可するといったポリシーを用いる。さらに、ポリシーをポリシーファイル２００に登録し、ファイルＩ／Ｏフックプログラム１０６がフックしたアクセス情報を、プロセス間通信プログラム１０８を介してセキュリティ用ＯＳ１０４上のアクセス制御プログラム１１０に渡し、前記ポリシーに基づいたアクセス制御を行う。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所